Ungati Ramesh* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.5, Issue No.2, February – March 2017, 5610-5614.

Implementations High Speed Of DSP Circuits Using Advanced Encryption Standard Algorithm

UNGATI RAMESH
Student
Sri Vaishnavi College of Engineering
(Approved by AICTE & Affiliated to JNTU, Kakinada), Singupuram (Vill & post), Srikakulam (Rural) Mandal.

DURGASI LOKESH
Asst. Professor
Sri Vaishnavi College of Engineering
(Approved by AICTE & Affiliated to JNTU, Kakinada), Singupuram (Vill & post), Srikakulam (Rural) Mandal.

Abstract: **The selective application of technological and related procedural safeguards is an important responsibility of every Federal organization in providing adequate security to its electronic data systems. This publication specifies a cryptographic algorithm, the Advanced Encryption Standard (AES) which may be used by Federal organizations to protect sensitive data. Protection of data during transmission or while in storage may be necessary to maintain the confidentiality and integrity of the information represented by the data.**

The algorithms uniquely define the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form. The Advanced Encryption Standard is being made available for use by Federal agencies within the context of a total security program consisting of physical security procedures, good information management practices, and computer system/network access controls. In this the key must be available at the transmitter and receiver simultaneously during communication. In this project both encryption and decryption process are implement. The functionality is verified using ISE simulator and the synthesis is carried out using XILINX ISE 12.3i.

Keywords: **AES; Encyption; Decryption; Xilinx ;**

I. INTRODUCTION TO AES

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not compare AES with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA.RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures.

AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key. AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e. an exhaustive search of all possible key combinations in order to find the right one.

In 1997 the US National Institute of Standards and Technology put out a call for candidates for a replacement for the ageing Data Encryption Standard, DES. 15 candidates were accepted for further consideration, and after a fully public process and three open international conferences, the number of candidates was reduced to five. In February 2001, the final candidate was announced and comments were solicited. 21 organizations and individuals submitted comments. None had any reservations about the suggested algorithm.

AES is founded on solid and well-published mathematical ground, and appears to resist all known attacks well. There's a strong indication that in fact no back-door or known weakness sexists since it has been published for a long time, has been the subject of intense scrutiny by researchers all over the world, and such enormous amounts of economic value and information is already successfully protected by AES. There are no unknown factors in its design, and it was developed by Belgian researchers in Belgium therefore voiding the conspiracy theories sometimes voiced concerning an encryption

standard developed by a United States government agency. A strong encryption algorithm need only meet only single main criteria:

- There must be no way to find the unencrypted clear text if the key is unknown, except brute force, i.e. to try all possible keys until the right one is found.

A secondary criterion must also be met:

- The number of possible keys must be so large that it is computationally infeasible to actually stage a successful brute force attack in short enough a time.

The older standard, DES or Data Encryption Standard, meets the first criterion, but no longer the secondary one – computer speeds have caught up with it, or soon will. AES meets both criteria in all of its variants: AES-128, AES-192 and AES-256.

II. ANALYSIS

AES may, as all algorithms, be used in different ways to perform encryption. Different methods are suitable for different situations. It is vital that the correct method is applied in the correct manner for each and every situation, or the result may well be insecure even if AES as such is secure. It is very easy to implement a system using AES as its encryption algorithm, but much more skill and experience is required to do it in the right way for a given situation. No more than a hammer and a saw will make anyone a good carpenter, will AES make a system secure by itself. To describe exactly how to apply AES for varying purposes is very much out of scope for this short introduction.

Encryption with AES is based on a secret key with 128, 192 or 256 bits. But if the key is easy to guess it doesn't matter if AES is secure, so it is as critically vital to use good and strong keys as it is to apply AES properly. Creating1 good and strong keys is a surprisingly difficult problem and requires careful design when done with a computer. The challenge is that computers are notoriously deterministic, but what is required of a good and strong key is the opposite – unpredictability and randomness. Keys derived into a fixed length suitable for the encryption algorithm from passwords or pass phrases typed by a human will seldom correspond to 128 bits much less 256. To even approach 128--bit equivalence in a pass phrase, at least 10 typical passwords of the kind frequently used in day-to-day work are needed. Weak keys can be somewhat strengthened by special techniques by adding computationally intensive steps which increase the amount of computation necessary to break it. The risks of incorrect usage, implementation and weak keys are in no way unique for AES; these are shared by all encryption algorithms. Provided that the implementation is correct, the security provided reduces to a relatively simple question about how many bits the chosen key, password or pass phrase really corresponds to. Unfortunately this estimate is somewhat difficult to calculate, when the key is not generated by a true random generator.

Security is not an absolute; it's a relation between time and cost. Any question about the security of encryption should be posed in terms of how long time, and how high cost will it take an attacker to find a key? Currently, there are speculations that military intelligence services possibly have the technical and economic means to attack keys equivalent to about 90 bits, although no civilian researcher has actually seen or reported of such a capability. Actual and demonstrated systems today, within the bounds of a commercial budget of about 1 million dollars can handle key lengths of about 70 bits. An aggressive estimate on the rate of technological progress is to assume that technology will double the speed of computing devices every year at an unchanged cost. If correct, 128-bit keys would be in theory be in range of a military budget within 30-40 years. An illustration of the current status for AES is given by the following example, where we assume an attacker with the capability to build or purchase a system that tries keys at the rate of one billion keys per second. This is at least 1 000 times faster than the fasted personal computer in 2004. Under this assumption, the attacker will need about 10 000 000 000 000 000 000 000 years to try all possible keys for the weakest version, AES-128. The key length should thus be chosen after deciding for how long security is required, and what the cost must be to brute force a secret key. In some military circumstances a few hours or days security is sufficient – after that the war or the mission is completed and the information uninteresting and without value. In other cases a lifetime may not be long enough.

III. PROPOSED MODEL

AES is a specification for the encryption of electronic data. It has been adopted by the U.S. government and is now used worldwide. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

AES operates on a 4×4 column-major order matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

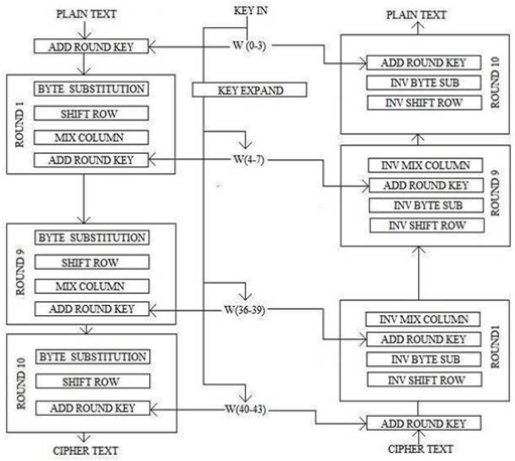


FIG 1 BLOCK DIAGRAM OF AES

The Sub bBytes step:

In the SubBytes step, each byte in the matrix is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

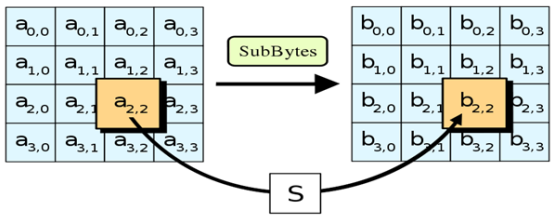


Fig 2 Sub bytes information

The ShiftRows step:

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

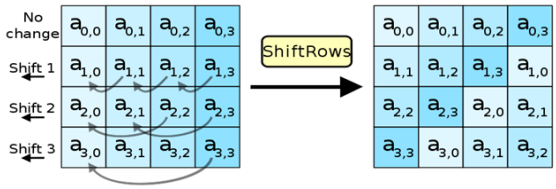


Fig 3 Shift rows information

The Mix Columns step:

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. TheMixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together withShiftRows, MixColumns provides diffusion i n the cipher.

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value. After shifting, a conditional xor with 0x1B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over **GF**(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from **GF**(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

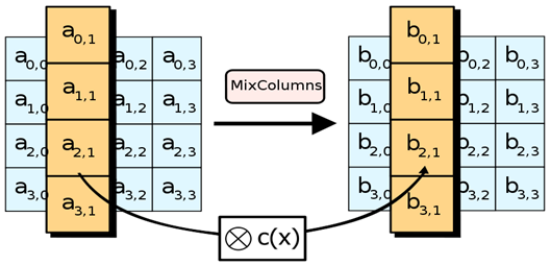


Fig 4 Mix Columns information

The Add RoundKey step:

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

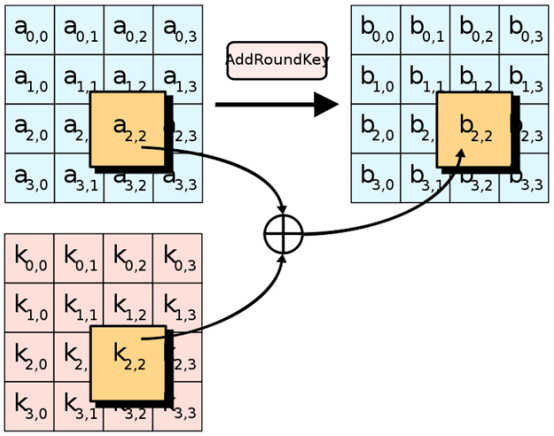


Fig 5 add around key information

Optimization of the cipher:

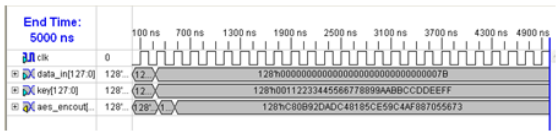
On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining SubBytes and ShiftRows withMixCol umns, and transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, which utilizes a total of four kilobytes (4096 bytes) of memory—one kilobyte for each table. A round can now be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step.

If the resulting four kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

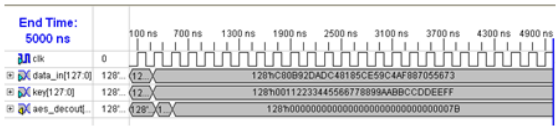
Using a byte-oriented approach, it is possible to combine the SubBytes, ShiftRows, and Mix Columns steps into a single round operation.

IV. RESULTS

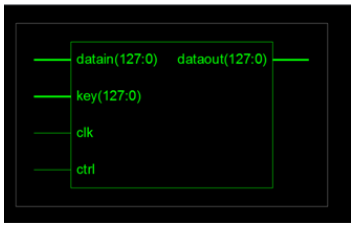
ENCRYPTION WAVEFORM



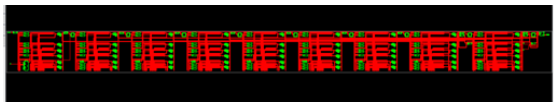
DECRYPTION WAVEFORM



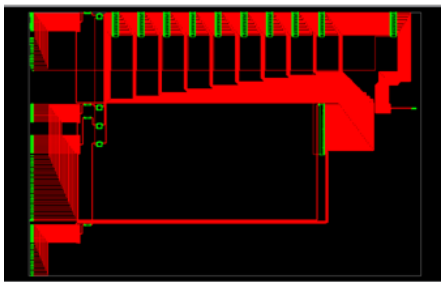
SCHEMATIC



RTL SCHEMATIC OF ENCRYPTION



RTL SCHEMATIC OF DECRYPTION



V. CONCLUSION

In this project, we have given 128 bits input and 128 bits security key and observed how it is delivered at the output with security. In this project there is no revealing of the original message to the hackers. The original message can be revealed to only sender and the receiver. So, in future, any proprierty information can be transmitted securely by using this project(military or banking purposes).

Modern applications of aes cover a wide variety of applications, such as secure internet (ssl), electronic financial transactions, remote access servers, cable modems, secure video surveillance and encrypted data storage.The future scope of our project is to extend 128 bits inputs to n bits(n is any integer value).

VI. REFERENCES

[1] J.Yang, J.Ding, N.Li and Y.X.Guo,"FPGA-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engin(CESCE).,Vol.02, Issue.5-6, pp.67-70, Jun 2010.

[2] A.M.Deshpande, M.S.Deshpande and D.N.Kayatanavar,"FPGA Implementation of AES Encryption and Decryption"IEEE Inter.Conf.Cont,Auto,Com,and Ener., vol.01,issue04, pp.1-6,Jun.2009.

[3] Hiremath.S. and Suma.M.S.,"Advanced Encryption Standard Implemented on FPGA" IEEE Inter.Conf. Comp Elec Engin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.

[4] Abdel-hafeez.S.,Sawalmeh.A. and Bataineh.S.,"High Performance AES Design using Pipelining Structure over GF(28)" IEEE Inter Conf.Signal Proc and Com.,vol.24-27, pp.716-719,Nov. 2007.

[5] Rizk.M.R.M. and Morsy, M., "Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA", IEEE Inter Conf. Desig Tes Wor.,vol.1,issue.16,pp.207-217, Dec. 2007.

[6] Liberatori.M.,Otero.F.,Bonadero.J.C. and Castineira.J. "AES-128 Cipher.High Speed, Low Cost FPGA Implementation", IEEE Conf. Southern Programmable Logic(SPL),vol.04,issue.07,pp.195-198,Jun. 2007.

[7] Abdelhalim.M.B., Aslan.H.K. and Farouk.H. "A design for an FPGAbased implementation of Rijndael cipher",ITICT. Ena TechSoc. (ETNKS), vol. 5, issue.6, pp. 897-912, Dec. 2005.

